

Seam Carving Algorithm

Κατά την μεταβολή των διαστάσεων μίας εικόνας, είναι μερικές φορές επιθυμητό η μεταβολή να γίνεται μόνο σε μία διάσταση. Για παράδειγμα, μπορεί να έχετε μία εικόνα από την οποία θέλετε να πάρετε μία στενότερη εικόνα, χωρίς να αλλάξει το ύψος της. Αντίστοιχα, μπορεί να έχετε μία εικόνα από την οποία θέλετε να πάρετε μία πιο κοντή εικόνα χωρίς να αλλάξει το πλάτος της.

Ο πιο απλός τρόπος για να αντιμετωπίσετε το πρόβλημα της μεταβολής σε μία διάσταση είναι είτε να κόψετε την εικόνα (*crop*), είτε να κλιμακώσετε την εικόνα (*scale*). Το κόψιμο της εικόνας έχει το μειονέκτημα ότι είναι πιθανόν να αφαιρέσετε και χρήσιμη πληροφορία, ενώ η κλιμάκωση σε μόνο μία διάσταση συμπιέζει την πληροφορία στην διάσταση αυτή παραμορφώνοντας την εικόνα.



Η αρχική εικόνα



Η διπλανή εικόνα έχει υποστεί κλιμάκωση στην οριζόντια διάσταση. Παρατηρήστε ότι η εικόνα έχει συμπιεστεί στην διάσταση αυτή, παραμορφώνοντας τα αντικείμενα που απεικονίζονται.



Η διπλανή εικόνα έχει υποστεί κόψιμο. Παρατηρήστε ότι από την αρχική εικόνα έχει χαθεί πληροφορία.

Σε αντίθεση με τα παραπάνω ο αλγόριθμος [Seam Carving](#) (Shai Avidan and Ariel Shamir, 2007) αλλάζει το μέγεθος της εικόνας σε μία μόνο διάσταση λαμβάνοντας υπόψη το περιεχόμενο της εικόνας. Ο αλγόριθμος καταλήγει σε καλύτερα αποτελέσματα για εικόνες των οποίων η πληροφορία είναι εστιασμένη σε συγκεκριμένα σημεία. Στην παραπάνω φωτογραφία, μη σημαντικά pixels όπως είναι ο ωκεανός και ο ουρανός αφαιρούνται από την εικόνα, ενώ pixels που περιέχουν πληροφορία, όπως είναι οι άνθρωποι και τα κύματα παραμένουν αναλλοίωτα.

Δείτε την παραπάνω εικόνα, έτσι όπως δημιουργείται μετά την εφαρμογή του αλγορίθμου *Seam Carving*. Παρατηρήστε ότι η πληροφορία που ενδιαφέρει (οι άνθρωποι, τα κύματα, το βουνό από πίσω) παραμένουν αναλλοίωτα, ενώ έχει αφαιρεθεί πληροφορία που δεν συνεισφέρει ουσιαστικά στην φωτογραφία.



Στην διπλανή εικόνα βλέπετε την εικόνα που δημιουργείται μετά την εφαρμογή του αλγορίθμου *Seam Carving*. Παρατηρήστε ότι η πληροφορία που ενδιαφέρει (οι άνθρωποι, τα κύματα, το βουνό από πίσω) παραμένουν αναλλοίωτα, ενώ έχει αφαιρεθεί πληροφορία που δεν συνεισφέρει ουσιαστικά στην φωτογραφία.

Το πλάτος της εικόνας έχει μειωθεί, ενώ το μήκος παραμένει αναλλοίωτο.

Επεξήγηση του αλγορίθμου

Υπολογισμός της ενέργειας της εικόνας

Ας υποθέσουμε ότι έχετε μία εικόνα H γραμμών και W στηλών ($H \times W$). Οι γραμμές της εικόνας αριθμούνται από μηδέν έως $H-1$, ενώ οι στήλες από 0 έως $W-1$. Για μία εικόνα 4 γραμμών και 3 στηλών ο πίνακας αρίθμησης για κάθε pixel (εικονοστοιχείο) δίνεται παρακάτω:

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)
(3,0)	(3,1)	(3,2)

Δημιουργούμε από την εικόνα ένα διδιάστατο πίνακα ιδίων διαστάσεων με την εικόνα που αποθηκεύει την “**ενέργεια**” της εικόνας. Η ενέργεια της εικόνας υπολογίζεται σε κάθε pixel (i,j) της γραμμής i και της στήλης j , ως εξής:

$$E(i,j) = E_x(i,j) + E_y(i,j),$$

$$E_x(i,j) = R_x(i,j)^2 + G_x(i,j)^2 + B_x(i,j)^2$$

$$E_x(i,j) = [R(i,j+1) - R(i,j-1)]^2 + [G(i,j+1) - G(i,j-1)]^2 + [B(i,j+1) - B(i,j-1)]^2$$

$$E_y(i,j) = R_y(i,j)^2 + G_y(i,j)^2 + B_y(i,j)^2$$

$$E_y(i,j) = [R(i+1,j) - R(i-1,j)]^2 + [G(i+1,j) - G(i-1,j)]^2 + [B(i+1,j) - B(i-1,j)]^2$$

όπου **$R(i,j)$** είναι η πληροφορία του κόκκινου χρώματος, **$G(i,j)$** είναι η πληροφορία του πράσινου χρώματος και **$B(i,j)$** είναι η πληροφορία του μπλε χρώματος για το pixel (i,j) .

Για τον υπολογισμό της ενέργειας στις ακμές της εικόνας, θεωρήστε ότι δεξιότερα των εικονοστοιχείων της στήλης $W-1$, βρίσκονται τα εικονοστοιχεία της στήλης 0 και αριστερότερα των εικονοστοιχείων της στήλης 0 βρίσκονται τα εικονοστοιχεία της στήλης $W-1$. Αντίστοιχα, θεωρήστε ότι μετά τα εικονοστοιχεία της τελευταίας σειράς της εικόνας (σειρά $H-1$) βρίσκονται τα εικονοστοιχεία της 1ης σειράς (σειρά 0) και πάνω από τα εικονοστοιχεία της 1ης σειράς βρίσκονται τα εικονοστοιχεία της τελευταίας σειράς.

(255, 101, 51)	(255, 101, 153)	(255, 101, 255)
(255,153,51)	(255,153,153)	(255,153,255)
(255,203,51)	(255,204,153)	(255,205,255)
(255,255,51)	(255,255,153)	(255,255,255)

Για την παραπάνω εικόνα, η ενέργεια του μεσαίου pixel (2,1) με τιμές **R=255**, **G=204**, **B=153** υπολογίζεται για την οριζόντια διάσταση από τα pixels (2,0) και (2,2).

$$\begin{aligned}
 R_x(1,2) &= 255 - 255 = 0, \\
 G_x(1,2) &= 205 - 203 = 2, \\
 B_x(1,2) &= 255 - 51 = 204 \\
 E_x(1,2) &= 2^2 + 204^2 = 41620
 \end{aligned}$$

και για την κάθετη διάσταση από τα pixels (1,1) και (1,3)

$$\begin{aligned}
 R_y(1,2) &= 255 - 255 = 0, \\
 G_y(1,2) &= 255 - 153 = 102, \\
 B_y(1,2) &= 153 - 153 = 0, \\
 E_y(1,2) &= 102^2 = 10404.
 \end{aligned}$$

Η συνολική ενέργεια για το pixel (1,2) είναι

$$E(1,2) = E_x(1,2) + E_y(1,2) = 41620 + 10404 = 52024$$

Αντίστοιχα, η ενέργεια για το ακραίο pixel (0,1) υπολογίζεται για την οριζόντια διάσταση από τα pixels (0,0) και (0,2)

$$\begin{aligned}
 R_x(0,1) &= 255 - 255 = 0, \\
 G_x(0,1) &= 101 - 101 = 0, \\
 B_x(0,1) &= 255 - 51 = 204, \\
 E_x(0,1) &= 204^2 = 41616
 \end{aligned}$$

και στην κάθετη διάσταση από τα pixels

$$\begin{aligned}
 R_y(0,1) &= 255 - 255 = 0, \\
 G_y(0,1) &= 255 - 153 = 102, \\
 B_y(0,1) &= 153 - 153 = 0, \\
 E_y(0,1) &= 102^2 = 10404.
 \end{aligned}$$

Η συνολική ενέργεια για το pixel (0,1) είναι

$$E(0,1) = E_x(0,1) + E_y(0,1) = 41616 + 10404 = 52020$$

Εντοπισμός οριζόντιας ή κατακόρυφης ραφής προς αφαίρεση

Αφού υπολογίσετε την ενέργεια της εικόνας το επόμενο βήμα είναι η αφαίρεση μιας ραφής (seam) από την εικόνα. Ας υποθέσουμε ότι θέλετε να μικρύνετε την εικόνα κατά την οριζόντια διάσταση. Σε αυτή την περίπτωση αναζητάτε την επόμενη κατακόρυφη ραφή προς αφαίρεση. Δείτε το παρακάτω παράδειγμα εντοπισμού της βέλτιστης ραφής

Ας υποθέσουμε ότι έχετε την παρακάτω εικόνα και τον πίνακα ενέργειας που την συνοδεύει:

(78,209, 79)	(63,118,247)	(92,175, 95)	(243, 73,183)	(210,109,104)	(252,101,119)
(224,191,182)	(108, 89, 82)	(80,196,230)	(112,156,180)	(176,178,120)	(142,151,142)
(117,189,149)	(171,231,153)	(149,164,168)	(107,119, 71)	(120,105,138)	(163,174,196)
(163,222,132)	(187,117,183)	(92,145, 69)	(158,143, 79)	(220, 75,222)	(189, 73,214)
(211,120,173)	(188,218,244)	(214,103, 68)	(163,166,246)	(79,125,246)	(211,201, 98)

57685.0	50893.0	91370.0	25418.0	33055.0	37246.0
15421.0	56334.0	22808.0	54796.0	11641.0	25496.0
12344.0	19236.0	52030.0	17708.0	44735.0	20663.0
17074.0	23678.0	30279.0	80663.0	37831.0	45595.0
32337.0	30796.0	4909.0	73334.0	40613.0	36556.0

Με βάση τον πίνακα της ενέργειας αναζητούμε την εκείνη την κάθετη ραφή της οποίας το άθροισμα της ενέργειας είναι το ελάχιστο. Η διαδρομή για κάθε κάθετη ραφή προκύπτει από τους εξής δύο περιορισμούς:

1. από το pixel (x,y) μπορούμε να μετακινηθούμε σε ένα από τα 3 pixels (x-1,y+1) ή (x,y+1) ή (x+1,y+1).
2. από τα τρία παραπάνω pixels μετακινούμαστε ΠΑΝΤΑ στο pixel με την χαμηλότερη ενέργεια.

Στο παραπάνω παράδειγμα η διαδρομή της ραφής ελάχιστης ενέργειας είναι η εξής: **[3, 4, 3, 2, 2]**

Η διαδρομή κάθε κάθετης ραφής ορίζεται μοναδικά από τους παραπάνω περιορισμούς. Επιπλέον, έχουμε τόσες δυνατές ραφές όσες είναι οι στήλες της εικόνας, καθώς κάθε κάθετη ραφή ξεκινάει από ένα pixel της 1ης γραμμής.

Η βέλτιστη ραφή για τον αλγόριθμο Seam Carving είναι εκείνη από τις W ραφές (για εικόνα με πλάτος W pixels) που έχει συνολικά την μικρότερη ενέργεια, όταν αθροίσουμε τις ενέργειες των pixels του μονοπατιού της. Επιλέγουμε να αφαιρέσουμε από την εικόνα την ραφή μικρότερης ενέργειας. Μετά την αφαίρεση, η εικόνα έχει μειωθεί κατά μία στήλη εικονοστοιχείων και πλέον έχει μέγεθος Hx(W-1), ενώ αρχικά είχε HxW. Επαναλαμβάνουμε την διαδικασία μέχρι η εικόνα να έρθει στο επιθυμητό μέγεθος εικονοστοιχείων στην οριζόντια διάσταση.

Προσοχή: Μετά την αφαίρεση μιας ραφής η εικόνα σας έχει αλλάξει, το ίδιο έχει αλλάξει και η ενέργεια της. Θα πρέπει να επανα-υπολογίσετε την ενέργεια πριν προχωρήσετε στον υπολογισμό της επόμενης ραφής.

Υλοποίηση σε Java

Δημιουργήστε την κλάση **ce325.hw1.SeamCarver** η οποία θα πρέπει να έχει κατ' ελάχιστο τις εξής μεθόδους.

```

public class SeamCarver {
    public SeamCarver(java.awt.image.BufferedImage image);
    public SeamCarver(java.io.File file);
    public SeamCarver(java.net.URL url);
    public double energy(int row, int col); // energy of a pixel
    public int[] findHorizontalSeam();      // return horizontal seam
    public int[] findVerticalSeam();        // return vertical seam
    public void removeHorizontalSeam(int[] seam); // remove the seam
    public void removeVerticalSeam(int[] seam);   // remove the seam
    private void scale(int width, int height);    // scale to the optimal
                                                    // dimensions before applying
                                                    // SeamCarve algorithm.
}
    
```



```
public void seamCarve(int width, int height); // Apply scale first
                                              // and seam carve afterwards.
}
```

Δείτε εδώ το σχετικό [javadoc](#).

Επιπλέον δημιουργήστε μία μέθοδο **main** η οποία υλοποιεί την διεπαφή με τον χρήστη ως εξής:

1. Εάν ο χρήστης δεν έχει δώσει το path της εικόνας από την γραμμή εντολών (URL ή file path) ζητά το path της εικόνας. Τότε ο χρήστης εισάγει το path. Εάν το path έχει εισαχθεί ως 1ο όρισμα κατά την κλήση του προγράμματος δεν απαιτείται το βήμα 1.
2. Ζητά τη νέα οριζόντια διάσταση της νέας εικόνας, ο χρήστης την εισάγει.
3. Ζητά τη νέα κάθετη διάσταση της νέας εικόνας, ο χρήστης την εισάγει.
4. Ζητά το όνομα του αρχείου στο οποίο θα αποθηκευτεί το αποτέλεσμα, ο χρήστης εισάγει μία εικόνα με κατάληξη **PNG**. Εάν το αρχείο υπάρχει το πρόγραμμα τερματίζεται χωρίς να γράψει τίποτα στο αρχείο.

Έλεγχος αποτελεσμάτων

Για κάθε εικόνα που κάνει μετατρέπεται το πρόγραμμα σας θα πρέπει σε ένα αρχείο με όνομα το όνομα της εικόνας και κατάληξη .dbg να κάνει τα εξής:

- Γράφει τις αρχικές διαστάσεις:
- Γράφει τις διαστάσεις μετά την εφαρμογή του αλγορίθμου κλιμάκωσης scale
- Γράφει σε μία γραμμή το path για κάθε ραφή που κάνει scale.

ΠΑΡΑΔΕΙΓΜΑ (TODO)

Τρόπος Αποστολής

Πακετάρετε **ΜΟΝΟ** τα αρχεία java αφού τα συμπιέσετε σε ένα αρχείο ZIP με όνομα το όνομα και το AEM κάθε μέλους της ομάδας σας (π.χ. *GiorgosThanos_1234_PeterGordon_1235.zip*). Τα αρχεία σας θα πρέπει να περιέχονται στην ιεραρχία καταλόγων που ορίζεται από το **package ce325.hw1**. Αποστέιλετε την δουλειά σας με e-mail στην διεύθυνση **ce325.course@gmail.com** ως εξής:

- Τίτλος (subject): **CE325 hw01**
- συνημμένο το παραπάνω αρχείο ZIP.
- Στο σώμα του μηνύματος τα ονόματα και τα AEM της ομάδας σας.

Εργασίες που δεν είναι συνεπείς με τους παραπάνω περιορισμούς δεν αξιολογούνται.